Memory Dump Page

Local Station Application

R. Goodwin

Nov 13, 1991

Function

This application is a venerable workhorse of Local Station diagnostics, maintenance and debugging. It displays 64 bytes of memory from either the local node or any set of nodes, updated at 15 Hz. This makes it possible to actually see what is happening in memory and thereby give a more complete understanding of the internal operation of system software. It allows setting memory words or bytes by keyboard entry. It can transfer memory blocks of any length from any location in any node to any other loca tion in any node. It can also fill a block of memory with any longword pattern, and it can dump a block of memory in standard S-record ascii format to any node's serial port.

Display layout and tour

```
M MEMORY DUMP     11/12/91 0814
0576:100000 0400 0010 0014 0000
0576:100008 0400 0040 0015 0000
0576:100010 0800 0004 0019 0000
0576:100018 0800 0010 0019 8000
0508:000788 9111 1208 1449 1120
0751:000788 9111 1208 1449 117F
0752:000788 9111 1208 1449 110D
0753:000788 9111 1208 1449 1114
ACCESS BY BYTES

MW 0576:00120000  0508:00120000
                       0000F000
MB      00000000  0576:FFFFA205
CACHE ON               00000001
```

This snapshot shows an example of the Memory Dump page application display. It shows 32 bytes of memory from the beginning of node 0508's system table directory, followed by 8 bytes of memory data from each of 4 nodes showing the current time-of-day. (Note that the time-of-day is well synchronized, as it is multicast on the network every minute by node 0516, which has access to a clock that is itself synchronized by reception of NBS satellite signals.)

Address entry

Up to 8 different addresses can be entered on the page. When an address is entered, that line and all of the following lines are automatically set to show consecutive memory data. So set different node/address combinations on different lines from the top down.

Bytes/words access mode

The data bytes presented on the page are accessed by bytes, which means that `MOVE.B` instructions are executed in the relevant contributing nodes to collect the data.

Another option is access by words, which causes `MOVE.W` instructions to be used instead. Normally one wouldn't expect there to be any difference, but some hardware boards are designed to be accessed in only one way. A keyboard interrupt in the first 18 columns of this line toggles between access by bytes and access by words.

Memory settings

One can enter new data to be stored at the memory locations displayed. The size of the setting can be one or two bytes, or it can be one word. To enter a word, type the new value over the data value that is currently displayed at that address. (The display will not overwrite while the cursor remains in the field.) With the cursor still positioned immediately to the right of the value just typed, press the keyboard interrupt button. Expect to see the new value displayed in place of the old one. The setting of the new value word will be set as two bytes if the display is using byte access and as a word if the display is using word access.

To make a single byte memory setting, type the two digits of the desired setting in place of the current byte value and use two spaces to cover up the other byte of the word on the display. Interrupt just after the word and only the single byte value will be set.

There is only the ability to set bytes and words manually. Other amounts of data can be set using the features described next. As an example, a longword setting could be set using the fill option, although the actual setting to the hardware would take place accessed by either two words or 4 bytes.

Copy, fill, and dump

The next four lines allow for copying memory blocks, filling them with constant data, or dumping them to a serial port. The first one is set to copy 60K bytes of memory words ("`MW`") from address $00120000 in node 0576 to the same address in node 0508. (This would copy the local station system program from node 0576 to node 0508.) The second one is set to copy a single 00 byte to the crate utility board 68901 chip data direction register. This results in disabling the heartbeat trigger, forcing the station to be reset by the watchdog timeout circuit on that board. The difference in specifying the fill option rather than the copy option is denoted by the absence of the colon that must be present in the first `nnnn:address` field. A keyboard interrupt action with the cursor in the area of either pair of lines will cause the indicated operation to be performed.

Other options available on these lines are `DB` and `DW`, which cause bytes or words of data from a memory block to be encoded into standard Motorola S-record ascii format and spooled to a serial port. To select the dump option, interrupt with the cursor in the first column, under the "`M`". Another interrupt switches back to the move/fill option. Byte or word access for these operations is similarly toggled by interrupt in the second column. The dump option provides a way of archiving memory data on the disk of a pc or host computer without using the network. Suggested future enhancements to these options might be to implement a find function and a verify function.

Cache enable

The last line displays the fact that the 68020 instruction cache is enabled in the local station. A keyboard interrupt on the left of this line toggles this enable.

Pointers

When examining software data structures, it is sometimes necessary to follow a pointer chain, or linked list. A facility for doing this is included on the Memory Dump page. An interrupt with the cursor located at either of the two middle hex digits of the upper word of a memory address that is currently displayed (as two words in the data area) causes the contiguous region displayed (usually 64 bytes) which includes that address to be replaced by the same size region starting at the address indicated. *It's much easier to do than to say.* Additional interrupts under addresses displayed can follow a linked chain of data structures. Each time this happens, the address (and node) is saved in an internal table. One can navigate through this table by interrupts with the cursor toward the right end of the line immediately following the 8 data lines, where an address will be shown. Interrupt under the left half of the address to back up in the table; interrupt in the right half to move forward through the saved addresses. If one backs up all the way to the start, the displayed address vanishes. Up to 100 addresses are saved internally, all of which is lost when leaving the page.

Freeze display

When the display is being updated at 15 Hz, it is sometimes difficult to catch values that change often, or to correlate them with other changing values. To assist in viewing a snapshot of the memory displayed, press the Hex button on the Local Console. The contents of the screen will be frozen as of that moment, and the "`NO UPDATE`" message will be displayed. Release the Hex button to resume normal updating. Note that while the display is frozen, the data is still being collected; it is simply not being updated on the display. Another way to make snapshots is via the serial port printing option below.

Volts units

If the data words being displayed are A/D readings, they can be shown in voltage units by selecting the Volts button on the Local Console. This assumes that the scaling used is ± 10 volts full scale. The value $8000 corresponds to –10 volts, and the value $7FFF represents +10 volts. Zero volts is $0000.

Entering addresses

To enter addresses in the memory data part of the display, there are several conventions in use. Typing an address alone at the start of a line, overwriting the node number, causes only the address part to be changed. But take care to terminate the address with a space, so that additional digits starting at the current cursor position are not accepted as part of the new address. If one merely changes the node digits, leaving the address alone, then only the node will be changed to match what has been modified. If both node and address are entered, both parts will be accepted.

The small local displays have only 32 characters per line, so there is not enough room to show a complete node#, a 32-bit address, and 4 data words with enough spacing to make it readable. Since it is common to use addresses within the 24-bit range, or to use node#s in which the hi byte is always the same, there is a toggle between two options of displaying the node:addresses. To alternate between the two options, interrupt in the first column of the second row—under the first character of the first node#.

If the interrupt is given with the cursor in the first column, then the address accepted is the address from the previous line but with the node number incremented by one. If there is no previous line, or if the cursor is on the last line, the current address is not changed; only the node number is incremented. And if the cursor is not on the last line, then the cursor is moved to the start of the next line. This makes it easy to look at the same 8 bytes of memory in successive nodes.

The raise and lower push buttons can be used to move forward and backward through memory. When the raise button is pressed, the address on the current line is advanced by the number of bytes of data that is displayed on that line plus the following lines, and the addresses are forced to be contiguous memory in the same node as on the cursor line. This allows one to scan through a region of memory displaying up to 64 bytes at a time. If the cursor is on a line *other* than one of the eight data lines, memory addresses are advanced 64 bytes at a time as if the cursor were on the first data line.

Smart Rack Monitor (SRM) access

Local stations connect to SRMs via Arcnet. A special range of node#s is reserved for this purpose. Arcnet nodes use node#s in the range `7AA0–7ABF`, allowing for a maximum of 32 SRMs on a single Arcnet network. New Linac control stations will typically have 3–4 SRMs accessible via Arcnet. Although the memory request protocol is different for SRMs than for local stations, special support is built into the memory dump page to permit viewing Arcnet-connected SRM memory in a natural way.

Small Memory Dump

This is a system function and is therefore available independent of what application page is running. Still, it affords one more line of 8 bytes that is available for display using the bottom line of the small screen display. (The easiest way to find it is to press the Home key and then the Up arrow key.) Enter an address at the start of the line; the node number is not required, as this line only displays local memory in bytes. If a bus error is found, the displayed data bytes are displayed in inverse video. Local memory settings can be made in the same way as is done via the Memory Dump page. The raise and lower push buttons can be used to move through local memory 8 bytes at a time while the cursor is on the bottom line.

Printing to the serial port

This is also a system function, but it bears including here. Any application page display can be written to the local serial port by a keyboard interrupt under the page title on the top line (actually the 3rd through the 18th column). Sometimes it's useful to save copies of the memory display for later perusal.

Error codes

In response to data requests, there can be an error code returned. The Memory Dump page displays this code on the top line in the second column (between the current page number and the page title) if it nonzero. Only a few codes are commonly seen. An **8** code means that some node whose data is requested in the list of addresses on the display is not returning any data in response to the request. This may mean that it's down or is off-line (not on the network), or the local station operating the memory dump page is off-line. A **7** code means the same thing, except that data was received from that node at least once since the request was last issued. This code appears often when nodes are not running in synchronization and is generally ignored. The other common code that may appear is a **4** code. It is described next.

Bus errors

The system has no means of predicting whether a given address from which data is requested will produce a bus error when it is accessed. When the access is made by the system code in processing the data request, it treads carefully. If a bus error is encountered while accessing the memory requested (by bytes or by words as selected) an error **4** is returned with the data in response to the request.

Internal matters

Memory copy block sizes

When moving memory blocks around, it is done with 1024-byte packets at 15 Hz. This means that 60K bytes of memory—which happens to be about the size of the system code—can be transferred in about 4 seconds. In the case of memory transfers to Arcnet-based SRMs, the maximum packet size is 480 bytes, in conformance with the smaller maximum fame size available with Arcnet.

SRM task names

Normal SRM communications are addressed to the network destination task name SRMD. But to download a new version of the SRM system code, the task name NEWV must be used. The memory dump page program checks for the special case of a new version download by the range of memory used in the memory copy. If it is in the range used for the system code, then it uses the NEWV task name; otherwise, it uses SRMD. After the transfer is complete, in the case that NEWV was used, a reset message is sent using the task name REST. This resets the SRM and causes it to begin execution of the new version.

Access to data from other nodes

Every data request or setting request includes a node number along with the memory address being referenced. The application program pays very little attention to whether the node number used is the local node or not. The system code that is invoked to make a data request does whatever is required to accomplish the task. If network communication is required, then it is used; otherwise, it is not. The job of writing application programs would likely be much more difficult if this were not the case. Again, to the user, this makes no difference at all. She needs only to specify the nodes she is interested in.

Command file

These modules are required to build the MDMP program:

SYSTEM	TRAP calls to system library routines
MDMP	Main Pascal object module
CACR	Library routine to access 68020 Cache Control Register

The program size is about 10K bytes.